



**Universitat Autònoma
de Barcelona**

Eticket: gestió d'esdeveniments i venda d'entrades online

Memòria del projecte

d'Enginyeria Tècnica en

Informàtica de Gestió

realitzat per

Mikel Petri Castro

i dirigit per

Vicenç Soler Ruíz

Escola d'Enginyeria

Sabadell, Setembre de 2011

El sotasignat, *Vicenç Soler Ruíz*,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció per

Mikel Petri Castro

I per a que consti firma la present.

Sabadell, *Setembre de 2011*

Signat: *Vicenç Soler Ruíz*

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

Títol del projecte: <i>Eticket: gestió d'esdeveniments i venda d'entrades online</i>	
Autor: Mikel Petri Castro	Data: Setembre de 2011
Tutor: Vicenç Soler Ruíz	
Titulació: Enginyeria tècnica en informàtica de Gestió	
Paraules clau <ul style="list-style-type: none">• Català: gestió, esdeveniments, venda, entrades, tiquets, facebook• Castellà: management, ventos, venta, entradas, tickets, facebook• Anglès: management, events, online, ticketing, tickets, facebook	
Resum del projecte <ul style="list-style-type: none">• Català: Avui dia, la gestió d'esdeveniments online i la seva promoció és un mercat creixent gràcies a l'evolució de la web 2.0 i a les xarxes socials. Ticketmaster és el referent, tot i així, no existeix una oferta clara per gestionar la venda d'entrades d'esdeveniments d'una mida petita o mitjana i que aprofiti les xarxes socials i la integració d'aplicacions que aquestes proporcionen. L'objectiu és realitzar una aplicació web on els usuaris (propis o de Facebook) puguin crear esdeveniments, publicar-los i promocionar-los a les xarxes socials, així com gestionar les tasques i característiques relacionades com la venda d'entrades i la llista de convidats.• Castellà: Hoy en día, la gestión de eventos online y su promoción es un mercado creciente gracias a la evolución de la web 2.0 y a las redes sociales. Ticketmaster es el referente, sin embargo, no existe una oferta clara para gestionar la venta de entradas de eventos de un tamaño pequeño o mediano y que aproveche las redes sociales y la integración de aplicaciones que éstas proporcionan. El objetivo es realizar una aplicación web donde los usuarios (propios o de Facebook) puedan crear eventos, publicarlos y promocionarlos en las redes sociales, así como gestionar las tareas y características relacionadas como la venta de entradas y la lista de invitados.• Anglès: Today, online event management and promotion is a growing market thanks to the evolution of Web 2.0 and to social networks. Ticketmaster is the reference, however, there is not a clear bid to manage the sale of tickets for small or medium size events and take advantage of social networks and the applications integration that these ones provide. The goal is to make a web application where users (own or Facebook ones) can create events, post and promote them in social networks, and also manage related tasks and features such as ticketing and guest list.	

Índex

1. Introducció	1
1.1. Objectius.....	1
1.2. Motivacions	1
1.3. Estat de l'art.....	1
1.4. Estructura de la memòria	2
2. Estudi de viabilitat	4
2.1. Introducció.....	4
2.2. Estudi de la situació actual	8
2.3. Requisits del projecte	9
2.4. Alternatives i selecció de la solució	11
2.5. Planificació del projecte	12
2.6. Avaluació de riscos	14
2.7. Pressupost	15
2.8. Conclusions.....	16
3. Fonaments teòrics	17
3.1. Servidor Web	17
3.2. Base de dades.....	18
3.3. Llenguatge <i>server-side</i>	18
3.4. Llenguatge <i>client-side</i> i AJAX	20
4. Anàlisi	22
4.1. Introducció.....	22
4.2. Tipus d'usuaris.....	22
4.3. Requeriments funcionals.....	23
4.4. Requeriments no funcionals.....	24
4.5. Anàlisi dels casos d'ús.....	25
4.6. Llei Orgànica de Protecció de Dades	28
5. Disseny.....	29
5.1. Estructura de la Base de Dades	29

5.2. Estructura de l'aplicació	31
5.3. Disseny de la interfície d'usuari.....	33
6. Implementació	36
6.1. Servidor WAMP	36
6.2. Implementació de l'aplicació.....	36
6.3. Implementació de la interfície d'usuari	40
6.4. Implementació de la connexió amb Facebook.....	42
7. Proves	46
7.1. Proves realitzades.....	46
8. Conclusions	47
8.1. Conclusions finals	47
8.2. Millores i ampliacions.....	48
9. Bibliografia	49
9.1. Llibres consultats	49
9.2. Pàgines web consultades	49
10. Apèndix	51
10.1. Índex de figures	51
10.2. Índex de taules	52

1. Introducció

1.1. Objectius

Aquest projecte tindrà com a objectiu el desenvolupar una aplicació web per a la gestió i venda d'entrades d'esdeveniments possibilitant la interacció amb la xarxa social més estesa, Facebook. L'aplicació permetrà als usuaris que es registrin o al que ja tinguin un compte a Facebook, crear esdeveniments i descriure'ls, vendre entrades, publicar l'esdeveniment a Facebook i veure el seu "mur", i invitar a gent mitjançant direccions de correu electrònic o "amics" de Facebook.

1.2. Motivacions

Les motivacions a l'hora d'escollir aquest projecte provenen principalment de la sorpresa que em va comportar observar la metodologia tan ineficient que s'utilitza a l'hora de vendre les entrades per les festes universitàries. La primera motivació és, per tant, la millora en l'eficiència de la venda i la distribució d'entrades, no tan sols en les festes universitàries sinó en qualsevol esdeveniment que no disposi del capital o el volum necessari per entrar en una gran web de distribució.

El segon motiu és la voluntat d'aprendre a desenvolupar una web dinàmica i integrada a les xarxes socials que pot oferir-me un coneixement sobre disseny web a tots els nivells (servidor, programació i disseny) molt útil de cara al futur laboral o a la posada en marxa d'un possible projecte empresarial propi.

1.3. Estat de l'art

Actualment existeixen alternatives disponibles per a la venda d'entrades online com *Eventbrite*, *Ticketleap* o *ticketea.com*. El problema és la poca integració que ofereixen amb les xarxes socials i la baixa disponibilitat de serveis fora d'Estats Units. Així doncs, s'optarà per desenvolupar un software propi que assoleixi els objectius esmentats.

1.4. Estructura de la memòria

La memòria consta de nou parts principals:

Introducció

Conté una presentació del projecte, objectius que es volen assolir, les motivacions que m'han portat a fer-lo i una petita nota sobre l'estat de l'art.

Estudi de viabilitat

S'exposa un estudi sobre la situació actual, els problemes existents, possibles solucions i alternatives, i una planificació pressupostària per determinar si el projecte és viable.

Fonament teòrics

S'hi pot trobar la definició de conceptes claus per desenvolupar el projecte així com una discussió sobre les diferents alternatives tecnològiques disponibles per realitzar el projecte.

Anàlisi

Es realitza l'anàlisi de requeriments necessari abans de realitzar qualsevol projecte.

Disseny

Inclou el disseny de la estructura interna de l'aplicació així com de la interfície d'usuari i la base de dades. Es fa un estudi per veure com s'haurien de dividir els mòduls i què hauria de fer cadascun d'ells.

Implementació

S'explica el procediment seguir per a materialitzar l'anàlisi i el disseny. Trobem com s'han implementat a la pràctica els mòduls, com es comunica l'aplicació amb Facebook, i com s'ha implementat la interfície d'usuari.

Conclusions

S'enumeren les conclusions objectives i personals a les que he arribat després d'haver realitzat el projecte.

Millores i ampliacions

Hi podem trobar una llista amb les diferents millores i ampliacions que m'han anat passant pel cap i no ha donat temps o eren massa difícils d'implementar. Algunes d'ells es podrien considerar un altre projecte

Bibliografia

Enumeració dels recursos bibliogràfics i electrònics consultats per a la realització del projecte.

2. Estudi de viabilitat

2.1. Introducció

2.1.1. Tipologia i paraules claus

Tipologia: Desenvolupament d'aplicació web.

Paraules claus: gestió, esdeveniments, venda, entrades, tiquets, tickets, facebook.

2.1.2. Descripció

La realització d'esdeveniments comporta normalment un afany de lucre i la venda d'entrades és el mitjà principal per garantir aquesta font d'ingressos. La organització per realitzar el procés de venda i de promoció comporta normalment uns costos en personal i recursos que es podrien estalviar si la gent compra el seu ticket a través d'internet. A través de les xarxes socials com Facebook pots promocionar els teus esdeveniments però no pots cobrar-ne entrada. Hi ha per tant, la necessitat de desenvolupar una eina que per poder realitzar aquestes tasques i optimitzar així el procés de gestió, promoció i distribució.

2.1.3. Objectius del projecte

1. Possibilitar als usuaris que es registrin i creïn esdeveniments.
2. Fer que els usuaris de Facebook puguin loguejar-se com un usuari propi.
3. Permetre la sincronització dels esdeveniments de Facebook i l'aplicació.
4. Crear entrades pels esdeveniments i vendre-les amb diferents mètodes de pagament.
5. Mostrar una llista dels propers esdeveniments.
6. Permetre als usuaris establir el lloc de l'esdeveniment mitjançant un mapa.
7. Buscar esdeveniments segons diferents característiques.
8. Invitar a persones als esdeveniments a través de direccions de correu electrònic o comptes de Facebook.

Priorització dels objectius

Objectius	Crític	Prioritari	Secundari
1	X		
2	X		
3	X		
4	X		
5		X	
6		X	
7			X
8			X

Taula 1. Prioritat dels objectius del projecte

2.1.4. Definicions, acrònims i abreviacions

PHP: *PHP Hypertext Pre-processor* – Llenguatge de programació web *server-side*.

AJAX: *Asynchronous JavaScript And XML* – Tècnica de programació web.

SDK: *Software Development Kit* – Conjunt d'eines pel desenvolupament d'aplicacions per a un sistema en concret.

JS: *JavaScript* – Llenguatge de programació web *client-side*.

API: *Application Programming Interface* – Conjunt de funcions i mètodes que ofereix certa llibreria a un altre software.

DOM: *Document Object Model* – API per representar, accedir i manipular objectes HTML.

HTML: *HyperText Markup Language* – Llenguatge per l'elaboració de pàgines web.

CSS: *Cascading Style Sheets* – Llenguatge per definir la presentació d'un document HTML.

Facebook: La xarxa social més popular i estesa.

Software: equipament lògic (aplicacions) d'una computadora.

MySQL: Sistema de gestió de bases de dades relacional de codi obert més estès.

Framework: Conjunt de mòduls de software que serveixen de base per altres projectes.

Plugin: Aplicació que es relaciona amb una altra per aportar-li una funció nova.

Widget: Petita aplicació per fer fàcil l'accés a funcions freqüents i oferir informació visual.

jQuery: JS Framework que simplifica la programació amb JS.

jQuery UI: Llibreria de components per jQuery que li afegeix plugins, widgets i efectes visuals.

Facebook Platform: Plataforma que ofereix un conjunt d'APIs i eines per interaccionar amb Facebook i els seus usuaris.

2.1.5. Parts interessades

Stakeholders		
Nom	Descripció	Responsabilitat
Vicenç Soler	Director del projecte	Supervisa la feina de l'alumne, avalua el projecte...
Mikel Petri	Usuari expert	Participa en la definició de requisits, representa a l'usuari tipus.

Taula 2. Stakeholders

Perfils d'usuari		
Nom	Perfil	Responsabilitat
Consumidor	Visitant	Comprar entrades.
Usuari	Usuari comú	Crea i assisteix a esdeveniments, ven i compra entrades.
Usuari Facebook	Usuari comú	Crea, assisteix a i sincronitza esdeveniments amb Facebook, ven i compra entrades.

Taula 3. Perfils d'usuari

Project Team		
Nom	Descripció	Responsabilitat
Mikel Petri	Cap del projecte (CP)	Defineix, gestiona, planifica i controla el projecte.
Mikel Petri	Analista (A)	Col·labora amb el cap de projectes en la redacció de la memòria. Analitza l'aplicació: arquitectura, estructura interna, especificacions, estàndards,... Participa en el disseny i validació.
Mikel Petri	Programador (P)	Desenvolupa l'aplicació d'acord amb l'anàlisi i planificació prevista. Participa en el procés d'implantació i proves.
Mikel Petri	Dissenyador (D)	Participa en el disseny de la interfície d'usuari de la web així com en la seva implementació
Mikel Petri	Becari (B)	Ajuda en el procés d'implementació i realitza les proves amb usuaris.
Vicenç Soler	Director del projecte (DP) i/o tutor	Supervisa la feina de l'alumne, en alguns casos també pot actuar com stakeholder.

Taula 4. Project Team

2.1.6. Referències

1. Normativa dels projectes d'enginyeries tècniques en informàtica.

http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf

2. LOPD: Llei Orgànica de Protecció de Dades.

https://www.agpd.es/portalwebAGPD/canaldocumentacion/legislacion/estatal/common/pdfs/Ley-15_99.pdf

3. Facebook Platform Policies: Normativa que aplica Facebook als seus usuaris i desenvolupadors.

<https://developers.facebook.com/policy/>

2.1.7. Producte i documentació del projecte

Es lliurarà una aplicació web i una memòria del projecte.

2.2. Estudi de la situació actual

2.2.1. Context

En la actualitat, la gestió d'esdeveniments es realitza a mà o en la pròpia web de l'organitzador, tot i així, la possibilitat per vendre entrades és reduïda al suport físic la gran majoria dels casos. Existeixen unes dificultats econòmiques pels que no són grans esdeveniments ja que s'ha de subcontractar algú per implementar la venda a la teua web o que et desenvolupi una plataforma per fer-ho si no en tens. Igualment, la falta de competència pel que fa la venda d'entrades dels grans esdeveniments és la culpable d'un cobrament de taxes abusiu. Pel que fa la promoció, avui dia pots promocionar els teus esdeveniments fàcilment amb Facebook però no facilita la personalització ni la integració amb una plataforma de pagaments.

2.2.2. Lògica del sistema

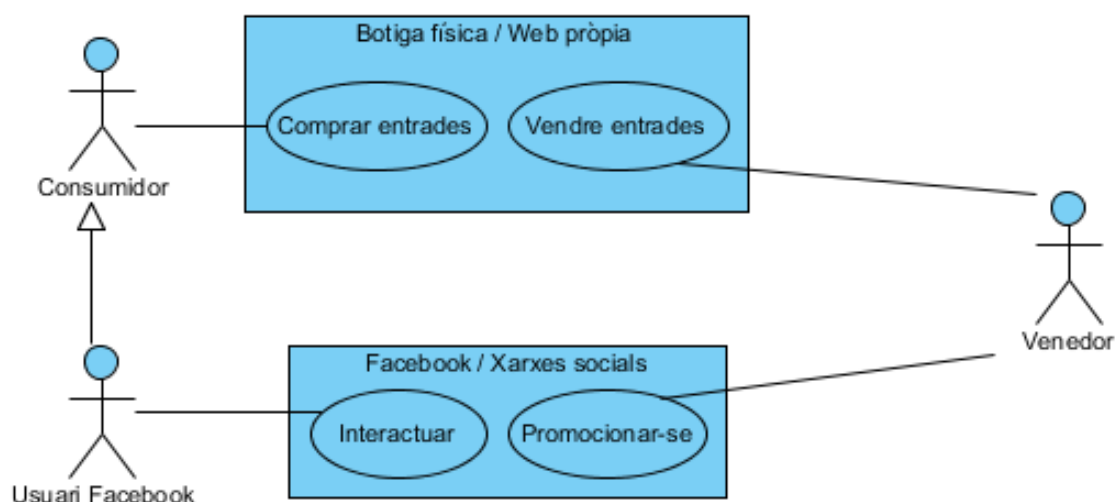


Figura 1. Lògica del sistema actual

2.2.3. Usuaris i/o personal del sistema

Nom	Perfil	Responsabilitat
Consumidor	Usuari comú	Comprar entrades.
Usuari de Facebook	Usuari comú	Interactua a les xarxes socials.
Venedor	Usuari expert	Ven les entrades i promociona els esdeveniments.

Taula 5. Usuaris i/o personal del sistema

2.2.4. Diagnòstic del sistema

Deficiències:

- Dedicació excessiva de recursos a la venda d'entrades.
- Incapacitat d'integrar els esdeveniments amb les xarxes socials.
- Dificultat alta a l'hora de vendre les entrades online.
- Incapacitat de vendre entrades per als esdeveniments no professionals.

Millores:

- Alliberament de recursos per vendre entrades.
- Integració i sincronització dels esdeveniments amb les xarxes socials.
- Facilitar la creació, gestió i promoció d'esdeveniments per als no professionals.
- Facilitar la decisió de vendre entrades online.

2.2.5. Normatives i legislació

- Normativa dels projectes de fi de carrera de les enginyeries tècniques. [1]
- Llei orgànica de protecció de dades. [2]
- Termes d'ús de la Facebook Platform. [3]

2.3. Requisits del projecte

2.3.1. Requisits funcionals

1. Crear esdeveniments amb els corresponents detalls.
2. Publicar i sincronitzar els esdeveniments a/amb les xarxes socials.
3. Comprar entrades pels esdeveniments.
4. Convidar a gent a l'esdeveniment.
5. Buscar esdeveniments.
6. Importar els esdeveniments de les xarxes socials.
7. Registrar-se i identificar-se.
8. Vendre entrades

2.3.2. Requisits no funcionals

1. Que l'aplicació tingui una resposta ràpida tot i un ús intensiu.
2. L'aplicació ha de tenir una aparença igual o gairebé igual independentment del navegador que s'utilitzi.
3. Els pagaments s'hauran de realitzar amb connexions xifrades.

2.3.3. Restriccions del sistema

- El projecte ha d'estar finalitzat abans del 20 de setembre de 2011.
- L'ordinador on s'implementarà el servidor de l'aplicació utilitza Windows 7.

2.3.4. Catalogació i prioritització dels requisits

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RNF1	RNF2	RNF3
Essencial	X	X	X				X	X	X		X
Condicional				X		X				X	
Opcional					X						

Taula 6. Catalogació i prioritització dels requisits

Objectius	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RNF1	RNF2	RNF3
1	X						X		X	X	
2							X		X	X	
3		X				X			X	X	
4			X					X	X	X	X
5					X				X	X	
6	X								X	X	
7					X				X	X	
8		X							X	X	

Taula 7. Relació entre els requisits i els objectius

2.4. Alternatives i selecció de la solució

2.4.1. Ticketmaster

Característiques:

- Líder en el mercat de *ticketing*.
- Múltiples mètodes de venda d'entrades.
- Preferiblement treballa amb esdeveniments grans.

Cost:

- Altes taxes que poden arribar al 50% del preu de l'entrada.
- Molt poca integració amb les xarxes socials.

2.4.2. Eventbrite o Ticketleap

Característiques:

- *Start-ups* amb una forta presència al mercat d'esdeveniments petits i mitjans.
- Diferents mètodes de pagament.
- Personalització dels esdeveniments.

Cost:

- Taxes mitjanes: 2,5% + 0,75€ [+ 3%] (procés de pagament) per entrada.
- Falta d'integració total a les xarxes socials.
- Pocs serveis fora d'Estat Units, aplicació en anglès.

2.4.3. Desenvolupar una aplicació pròpia

Característiques:

- Gran integració amb les xarxes socials.
- Localització local dels serveis.
- Taxes molt competitives.
- Diferents plataformes de pagament.

Cost:

- Segons la planificació prevista

2.4.4. Solució proposada

Alter.	Taxes	Mètodes de pagament	Integració xarxes socials	Esdeveniments tipus	Idioma
1	Molt altes	Molts	Baixa	Grans	Castellà
2	Mitjanes	Molts	Mitjana	Petit/Mitjà	Anglès
3	Baixes	Molts	Alta	Petit/Mitjà	Castellà

Taula 8. Comparació d'alternatives

Comparant les diferents característiques s'ha decidit triar **l'alternativa 3** perquè és la que més s'ajusta als nostres requisits. Els costos estaran determinats per la planificació següent.

2.5. Planificació del projecte

Calendari del projecte: El projecte és desenvoluparà durant l'agost i part de setembre del 2011 amb una dedicació mitjana d'unes 6 hores diàries. El total d'hores dedicades al projecte serà de 300 hores.

Data de començament: 1 d'agost de 2011.

Data finalització: 19 de setembre de 2011.

Eines de planificació i control: Microsoft Project 2010.

2.5.1. Recursos del projecte

Recursos humans	Valoració	Recursos materials: S'utilitzaran els recursos materials del cap de projecte. Tot el desenvolupament es farà utilitzant programari lliure o sense cost. Costos indirectes: amortització dels recursos de desenvolupament.
Cap de projecte	60 €/h	
Analista	30 €/h	
Programador	17 €/h	
Dissenyador	18 €/h	
Becari	6 €/h	

Taula 9. Recursos del projecte

2.5.2. Tasques del projecte

	Nombre de tarea	Duración	Prede	Nombres de los recursos
1	Eticket: gestió d'esdeveniments i venda d'entrades online	50 días		
2	Estudi de Viabilitat	20 hrs		Cap del projecte
3	Estudi dels Fonaments Teòrics	5,67 días	2	
4	Base de dades MySQL	8 hrs		Becari
5	Servidor Apache	8 hrs	4	Becari
6	PHP	12 hrs	5	Becari
7	jQuery	6 hrs	6	Becari
8	Anàlisi	22 hrs	3	Analista
9	Disseny	5,67 días	8	
10	Estructura BBDD	4 hrs		Analista;Programador
11	Estructura de l'aplicació	15 hrs	10	Analista;Programador
12	Disseny de la interfície	15 hrs	11	Dissenyador;Analista
13	Implementació	22,33 días	9	
14	Implementació del servidor WAMP	4 hrs		Programador;Becari
15	Implementació de la aplicació	100 hrs	14	Programador[80%];Becari[20%]
16	Implementació del disseny	65 hrs	14;15FF	Dissenyador;Programador[20%]
17	Implementació de la connexió amb Facebook	30 hrs	15;16	Programador;Becari
18	Proves	1,5 días	13	
19	Proves del programadors	4 hrs		Programador
20	Proves dels usuaris	5 hrs	19	Becari
21	Realització de la memòria	47 hrs	18	Cap del projecte;Analista

Figura 2. Tasques del projecte detallades

2.5.3. Planificació temporal

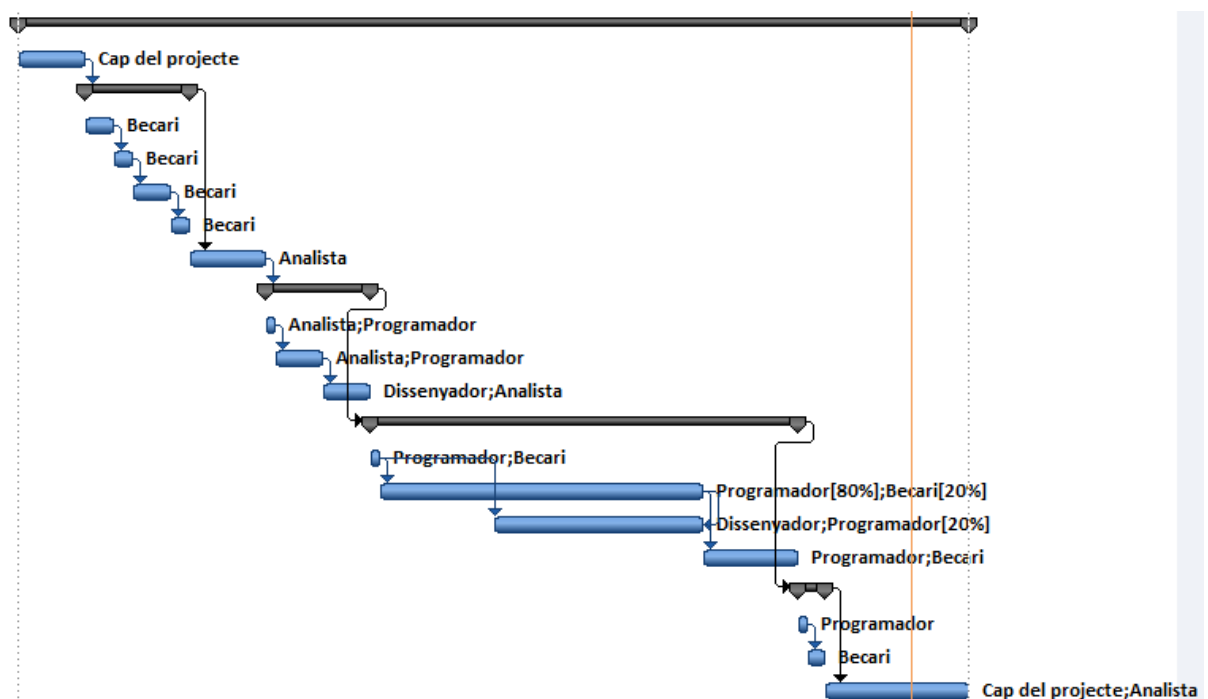


Figura 3. Diagrama de Gantt previst

2.6. Avaluació de riscos

2.6.1. Llista de riscos

1. Incapacitat de la aplicació per ser *cross-browser*¹. Implementació.
2. Diferències en la interfície gràfica depenent del navegador. Implementació.
3. Seguretat de l'aplicació. Mantenir la privacitat que els usuaris estableixin. Disseny.
4. Tecnologia triada inadequada. Que no s'adapti als requisits de l'aplicació. Implementació

2.6.2. Catalogació de riscos

Risc	Probabilitat	Crític
1	Baixa	Crític
2	Alta	Marginal
3	Mitjà	Crític
4	Baixa	Molt Crític

Taula 10. Riscos del projecte

2.6.3. Pla de contingència

Risc	Solució que cal adoptar
1	Realitzar els canvis convenients al codi perquè suporti tots els navegadors.
2	Implementar tots els estàndards i introduir codi condicionat al navegador que s'empri (principalment: IE vs Firefox/Chrome).
3	Aplicar polítiques de seguretat d'accés restringit a les dades.
4	Intentar aplicar altres tecnologies als mòduls problemàtics que no s'ajusten als requeriments. Possible reprogramació del mòdul.

Taula 11. Plans de contingència

¹ Capacitat d'una aplicació web per suportar tots els navegadors.

2.7. Pressupost

2.7.1. Estimació cost de personal

Nom del recurs	Treball	Cost
Cap del projecte	67 h	4.020,00 €
Analista	103 h	3.090,00 €
Programador	151,2 h	2.570,40 €
Dissenyador	80 h	1.440,00 €
Becari	93 h	558,00 €

Taula 12. Estimació cost de personal

Total: 11.678,40€

2.7.2 Estimació cost dels recursos

	Cost amortització	Cost unitari	Període amortització	Període utilització
Amortització PC programador	50 €	1200 €	48 mesos	2 mesos
Amortització MS Office	13,89 €	250 €	36 mesos	2 mesos
Amortització MS Project	20 €	360 €	36 mesos	2 mesos

Taula 13. Estimació cost dels recursos

2.7.3. Resum i anàlisi cost benefici

Cost del projecte.....	11.678,40 €
Cost d'amortització del material.....	83,89 €
Total:	11.762,29 €

2.8. Conclusions

2.8.1. Beneficis

- Facilitat de creació d'esdeveniments online.
- Facilitat a l'hora de sincronitzar-los amb les xarxes socials.
- Una plataforma amb molts esdeveniments on és fàcil buscar-los i promocionar-los.
- Permet externalitzar el procés de venda d'entrades a la xarxa.

2.8.2. Inconvenients

- Recel per part dels consumidors a comprar per Internet.
- Cobrament d'unes taxes que podrien evitar-se si es venguessin les entrades físicament.

Un cop valorats els beneficis i els inconvenients es considera que aquest projecte és viable en termes de costos, temps i beneficis futurs.

3. Fonaments teòrics

Seguidament s'explicaran alguns conceptes teòrics relacionats amb el projecte així com les tecnologies escollides per la seva realització enumerant els motius que s'han tingut en compte a l'hora de triar-les.

3.1. Servidor Web

Un servidor Web és una aplicació que segueix el model client-servidor i s'encarrega de esperar peticions de recursos per servir-los mitjançant algun tipus de protocol, generalment, HTTP. Els recursos poden ser estàtics –el servidor només s'encarrega de buscar el recurs i enviar-lo– o dinàmics –el servidor s'encarrega d'executar un script d'un llenguatge determinat per cada consulta generant així el recurs– Aquestes peticions de recursos es realitzen mitjançant una seqüència de caràcters i estandarditzada anomenada URL (*Uniform Resource Locator*). S'assigna una URL única a cada recurs d'informació disponible a Internet. Per solucionar el problema de HTTP, que no permet guardar informació sobre connexions anteriors, van aparèixer les *cookies*, on es pot mantenir informació d'interès pel servidor al sistema del client.

Es van estudiar diverses opcions:

- **nginx:** És un servidor lleuger d'alt rendiment de codi obert i multiplataforma. Va ser desenvolupat per les necessitats d'alguns webs d'atendre un gran número de peticions.
- **Apache:** Servidor web de codi obert i multiplataforma. És el servidor web més utilitzat¹ de disseny modular i amb una gran quantitat de documentació i una comunitat molt extensa.
- **lighttpd:** És un servidor web molt optimitzat per servir en entorns on la velocitat és clau. Una de les seves característiques principals és el seu baix consum de recursos en comparació amb altres servidors. Oficialment només funciona en Linux.

Es va decidir optar per Apache degut a la seva popularitat i documentació per poder solucionar cap problema ràpidament. També es va tenir en compte el ser multiplataforma ja que l'entorn de desenvolupament és en Windows 7.

¹ Segons la Netcraft Web Server Survey <http://news.netcraft.com/archives/2011/08/05/august-2011-web-server-survey-3.html>

3.2. Base de dades

Una base de dades és un conjunt de dades emmagatzemades per al seu ús posterior. Qui s'encarrega d'emmagatzemar, accedir, mantenir i estructurar les dades és el DBMS (***D**atabase **M**anagement **S**ystem*). El model més utilitzat avui dia per implementar bases de dades és el model relacional, que permet establir interconnexions entre les dades emmagatzemades en taules per establir relacions i restriccions. Les taules estan formades per camps/columnes i registres/files. Cada registre és identificat pels valors que té de cada camp. Una taula té un número de camps determinat i un nombre qualsevol de registres. Per accedir a les dades emmagatzemades en BBDD relacionals s'utilitza un llenguatge declaratiu que permet especificar diversos tipus d'operacions anomenat SQL (***S**tructured **Q**uery **L**anguage*). SQL permet realitzar tant consultes per recuperar les dades com actualitzacions o insercions.

Les opcions estudiades van ser:

- **MySQL:** RDBMS multifil i multiusuari. És el sistema gratuït més popular i estès. El codi i l'aplicació estan disponibles sota la llicència GNU General Public License, tot i que pertanyen a la companyia MySQL AB (Oracle). S'utilitza tant per projectes petits com per gegants com Google, Facebook, Wikipedia, YouTube...
- **MS Acces:** RDBMS orientat a un ús personal o de petites organitzacions. La seva escalabilitat és molt pobre així com la seva velocitat de resposta i rendiment
- **Oracle / MS SQL Server:** Capacitats per grans projectes però amb un alt cost econòmic i alt consum de recursos computacionals i de memòria.

Tenint en compte les diferents característiques es va decidir escollir MySQL per la seva documentació, popularitat, facilitat d'instal·lació, rendiment i gratuïtat.

3.3. Llenguatge *server-side*

Tal com hem dit abans, els continguts web poden ser dinàmics o estàtics. Per tal de generar webs interactives i amb una experiència rica per l'usuari, les pàgines webs s'han de generar dinàmicament tenint en compte la identitat, les dades disponibles, les accions anteriorment realitzades, etc.

Existeixen llenguatges que fan possible la generació de continguts sota demanda mitjançant l'execució de scripts.

Primerament, l'autor de l'aplicació crea el script que serà servit. Quan és produïda una petició, el servidor busca el fitxer i se'l passa al motor de scripting perquè analitzi i interpreti el contingut per generar el recurs que es demana. Un cop generat s'envia el recurs (normalment codi HTML) al client i aquest el processa i mostra en pantalla (en el cas d'una petició web).

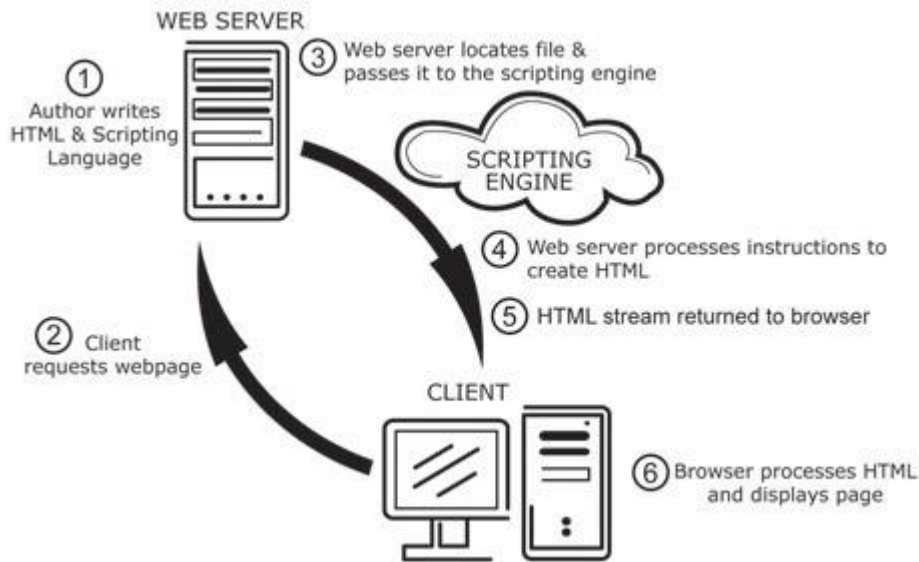


Figura 4. Esquema de generació dinàmica de continguts

Les diferents alternatives que es van tenir en compte varen ser:

- **PHP**: Llenguatge distribuït sota la PHP License, gratuït, multiplataforma, i àmpliament difós a tot el món. S'utilitza a una gran varietat de projectes com Moodle, Wikipedia o Facebook. Disposa de moltíssima documentació i una gran integració amb altres aplicacions a través de SDKs. Molt semblant a C.
- **ASP**: Llenguatge propietat de Microsoft i no és lliure. El seu ús és molt qüestionat ja que només funciona IIS (Servidor Web de Windows).
- **JSP**: Llenguatge propietat de Sun Microsystems, multiplataforma i gratuït. Permet executar codi Java per generar continguts web.

Analitzant els pros i els contres, es va decidir triar PHP per la seva estesa documentació, la integració que ofereix amb altres plataformes a través de SDKs, i la experiència prèvia que disposa l'estudiant en aquest llenguatge minimitzant el temps d'aprenentatge.

3.4. Llenguatge *client-side* i AJAX

3.4.1. JavaScript

JavaScript (d'ara en endavant, JS) és el llenguatge de scripting del costat del client per excel·lència ja que tots els navegadors majoritaris el suporten. És un llenguatge semblant a C i amb característiques de Java (tot i no tenir res a veure amb el ell, malgrat el nom). El codi JS s'insereix dins del codi HTML amb l'etiqueta `<script>` i el navegador l'executa mitjançant un motor de scripting a l'ordinador del client. És una eina bàsica per implementar dinamisme introduint interacció entre la pàgina i l'usuari sense que aquest hagi de recarregar la pàgina. JS accedeix al codi HTML i interactua amb ell mitjançant una implementació del DOM (**Document Object Model**), una API que proporciona un conjunt estàndard d'objectes per representar documents HTML i XML desenvolupada pel World Wide Web Consortium (W3C).

3.4.2. AJAX

AJAX és una model de desenvolupament web que permet actualitzar continguts web sense tornar a carregar la pàgina, és a dir, asíncronament, permetent crear pàgines altament interactives. AJAX utilitza una combinació de diverses tecnologies per actualitzar la web: l'objecte *XMLHttpRequest* és cridat utilitzant JS i aquest s'encarrega de realitzar la petició (sense interferir en cap moment amb l'aparença de la web) que, un cop resposta, permet a JS i l'accés a la plana a través del DOM actualitzar el contingut desitjat.

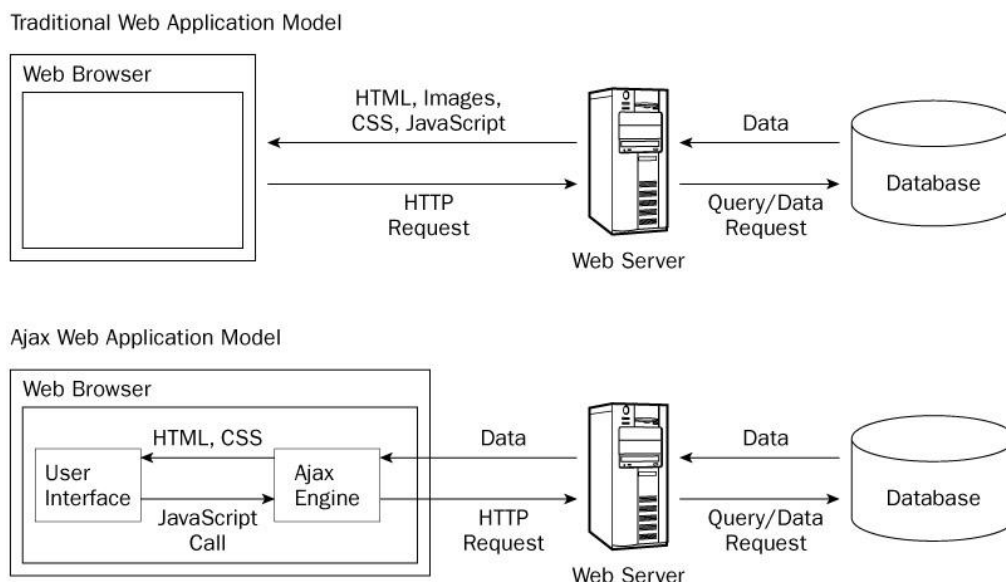


Figura 5. Esquema de funcionament del model AJAX

La resposta sol ser en format XML o JSON (*JavaScript **Object Notation***), el més comú. JSON és un estàndard basat en text pla i dissenyat per a l'intercanvi de dades (objectes) llegible pels humans.

Tot i els avantatges, AJAX també presenta inconvenients e.g. augmenta la dificultat de desenvolupament de les pàgines i no guarda les peticions a l'historial del navegador i el botó d'enrere perd la seva utilitat.

3.4.3. jQuery i jQuery UI

Per tal de facilitar la feina de programar JS existeixen diferents *frameworks* o llibreries que simplifiquen notablement certes tasques. Alguns exemples són:

- Prototype juntament amb Script.aculo.us (Efectes visuals)
- YUI
- jQuery juntament amb jQuery UI (Efectes visuals)

No hi ha diferències molt destacables entre elles a nivell de compatibilitat ja que treballen sobre JS. Es va decidir triar jQuery ja que disposa d'una gran acceptació i documentació a la xarxa, un gran nombre de *plugins* independents i uns *widgets* molt útils a través de jQuery UI.

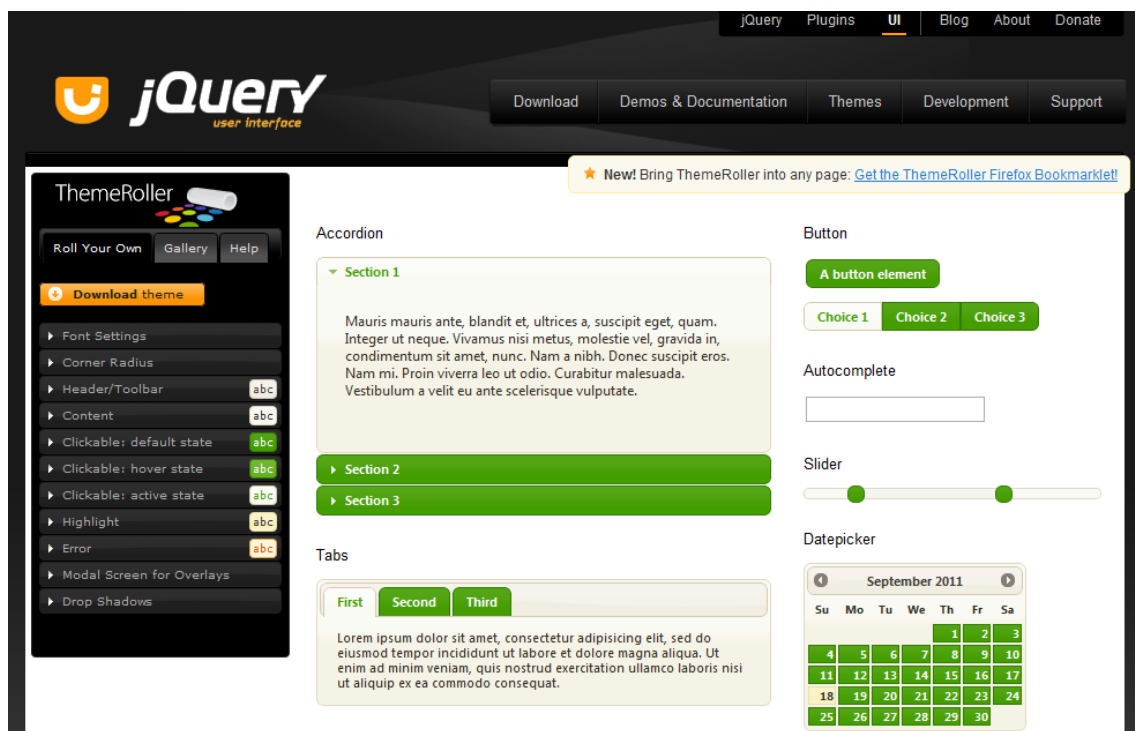


Figura 6. Widgets jQuery UI

4. Anàlisi

4.1. Introducció

Seguidament s'explicaran les característiques i requeriments que haurà de tenir el projecte així com els diferents tipus d'usuaris i com interactuaran amb el sistema amb els casos d'ús corresponents. Finalment s'inclourà una secció sobre les implicacions que la LOPD té sobre el projecte.

Funcions de l'aplicació:

- Registrar-se
- Login propi i amb Facebook
- Crear event
- Buscar event
- Publicar event a Facebook
- Vendre entrades

4.2. Tipus d'usuaris

Hi hauran 3 tipus d'usuaris segons si han accedit amb la seva compte d'usuari, amb la de Facebook, o si no han accedit amb cap.

4.2.1. Usuari sense sessió iniciada

Es tracta de l'usuari per defecte. Tots els clients que visiten la pàgina són usuaris sense sessió i poden comprar entrades si l'esdeveniment és públic. Poden buscar i visualitzar esdeveniments públics.

4.2.2. Usuari propi

Són els usuaris registrats a l'aplicació que han utilitzat les seves credencials per accedir-hi. Són els usuaris que volen utilitzar l'aplicació sense les opcions que ofereix la interacció amb Facebook. Poden crear, buscar i visualitzar esdeveniments (si són públics o estan convidats), comprar o vendre entrades, convidar a gent als seus esdeveniments via direcció de correu i vincular-se amb el seu compte de Facebook. Si un usuari propi vincula el seu compte amb el de Facebook només podrà tornar accedir amb el login de Facebook.

4.2.3. Usuari Facebook

És l'usuari complet, té les funcionalitats de l'usuari identificat més l'opció de publicar els esdeveniments a Facebook, mantenir-los sincronitzats i convidar a amics del seu compte. Un usuari de Facebook també pot revocar els permisos a l'aplicació si ho desitja recuperant així el seu compte d'usuari propi. Els esdeveniments públics recopilats a través del seu compte no podran ser eliminats ja que són públics.

4.3. Requeriments funcionals

4.3.1. Usuari sense sessió iniciada

- Comprar entrades
- Buscar esdeveniments públics
- Visualitzar esdeveniments públics

4.3.2. Usuari propi

- Comprar entrades
- Buscar esdeveniments
- Visualitzar esdeveniments
- Convidar a gent
- Crear esdeveniments
- Vendre entrades
- Vincular compte amb la de Facebook

4.3.3. Usuari Facebook

- Comprar entrades
- Buscar esdeveniments
- Visualitzar esdeveniments
- Convidar a gent als esdeveniments via Facebook
- Crear esdeveniments
- Vendre entrades

- Publicar els esdeveniments a Facebook
- Sincronitzar amb els esdeveniments de Facebook
- Revocar els permisos a l'aplicació.

4.4. Requeriments no funcionals

Que l'aplicació tingui una resposta ràpida tot i un ús intensiu:

L'aplicació ha de donar una resposta adequada tot i que hi hagin pics de demanda. S'ha de garantir que els scripts estaran optimitzats i que els documents relacionats amb les pàgines com fitxers JS es carregaran asíncronament quan sigui possible.

L'aplicació ha de tenir una aparença igual o gairebé igual independentment del navegador que s'utilitzi:

La web canvia constantment i sovint s'afegeixen nous estàndards o s'implementen funcions de versions posteriors encara que l'estàndard no estigui acabat, com amb HTML5 i CSS 3. Es faran proves amb les últimes versions dels tres navegadors majoritaris: IE, Firefox i Chrome per tal de cobrir un 70% del total del mercat aproximadament.¹ També es realitzaran algunes proves amb navegadors antics per tal d'aconseguir una visualització el més semblant possible a la dels últims navegadors.

Els pagaments s'hauran de realitzar amb connexions xifrades:

La recollida de dades de la targeta i el procés de pagament s'haurà de realitzar sota el protocol HTTPS (HTTP + SSL/TLS), basat en HTTP però destinat a la transferència segura de dades mitjançant el xifrat del canal de transmissió. Aconsegueix crear un canal segur sobre una xarxa insegura.

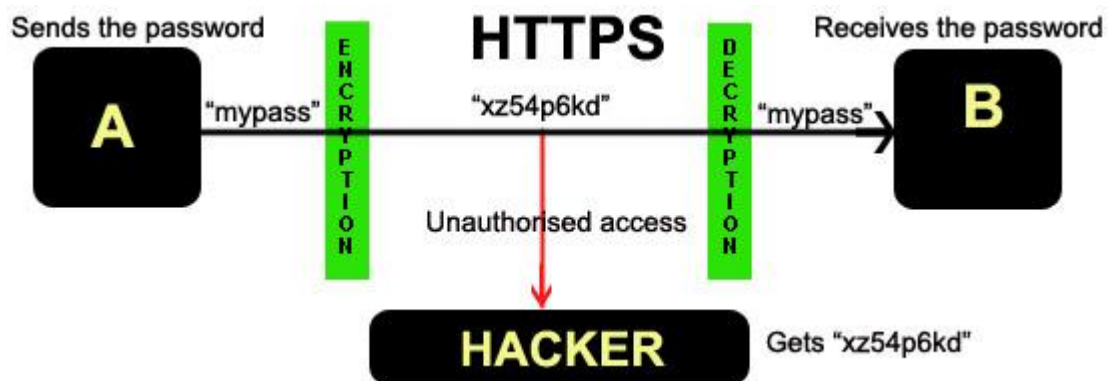


Figura 7. Esquema del protocol HTTPS

¹ http://www.w3schools.com/browsers/browsers_stats.asp
(Estadístiques individuals per cada navegador i versió)

4.5. Anàlisi dels casos d'ús

4.5.1. Descripció dels casos d'ús

Comprar entrades:

- **Descripció:** Els usuaris podran comprar entrades mitjançant diverses plataformes de pagament.
- **Usuaris:** Usuari sense sessió iniciada, Usuari propi i Usuari Facebook.

Buscar esdeveniments *include* Buscar esdeveniments públics:

- **Descripció:** Es podran buscar esdeveniments especificant el títol. Si no són públics només es mostraran als convidats.
- **Usuaris:** Usuari sense sessió iniciada (només públics), Usuari propi i Usuari Facebook.

Visualitzar esdeveniments *include* Visualitzar esdeveniments públics:

- **Descripció:** Veure la web del esdeveniment amb els seus detalls corresponents. Si no són públics només podran veure-la els seus convidats.
- **Usuaris:** Usuari sense sessió iniciada (només públics), Usuari propi i Usuari Facebook.

Crear esdeveniments:

- **Descripció:** Crear un esdeveniment incloent els seus detalls i.e. títol, fotografia, ubicació, descripció, privacitat...
- **Usuaris:** Usuari propi i Usuari Facebook.

Convidar a gent:

- **Descripció:** Es pot convidar a gent a qualsevol esdeveniment via e-mail. Els convidats seran els únics que puguin visualitzar l'esdeveniment si aquest és privat.
- **Usuaris:** Usuari propi i Usuari Facebook.

Vendre entrades:

- **Descripció:** Es podran crear diferents entrades amb quantitats limitades i diferents preus per vendre-les aplicant una petita taxa.
- **Usuaris:** Usuari propi i Usuari Facebook.

Vincula compte amb el de Facebook:

- **Descripció:** Es podrà optar per transformar el teu compte d'usuari propi en una d'Usuari Facebook mitjançant un vincle entre totes dues.
- **Usuaris:** Usuari propi.

Convidar a gent als esdeveniments via Facebook *extends* Convidar a gent:

- **Descripció:** Es pot convidar a gent a qualsevol esdeveniment via e-mail o notificació als amics de Facebook. Els convidats seran els únics que puguin visualitzar l'esdeveniment si aquest és privat.
- **Usuaris:** Usuari Facebook.

Publicar els esdeveniments a Facebook

- **Descripció:** Els esdeveniments de l'aplicació es podran publicar a Facebook per facilitar la promoció i aprofitar les característiques pròpies com el mur.
- **Usuaris:** Usuari Facebook.

Sincronitzar els esdeveniments amb Facebook

- **Descripció:** Els esdeveniments de l'aplicació es podran publicar i mantenir-se sincronitzats si es produeix algun canvi en qualsevol dels dos.
- **Usuaris:** Usuari Facebook.

Revocar els permisos a l'aplicació:

- **Descripció:** Els permisos que necessita l'aplicació per mantenir la interacció amb Facebook poden ser revocats per tornar a un compte d'usuari propi o simplement per desvincular l'aplicació del compte Facebook.
- **Usuaris:** Usuari Facebook.

4.5.2. Diagrama de casos d'ús

Seguidament es mostrarà el diagrama de casos d'ús de l'aplicació interactuant amb els seus usuaris.

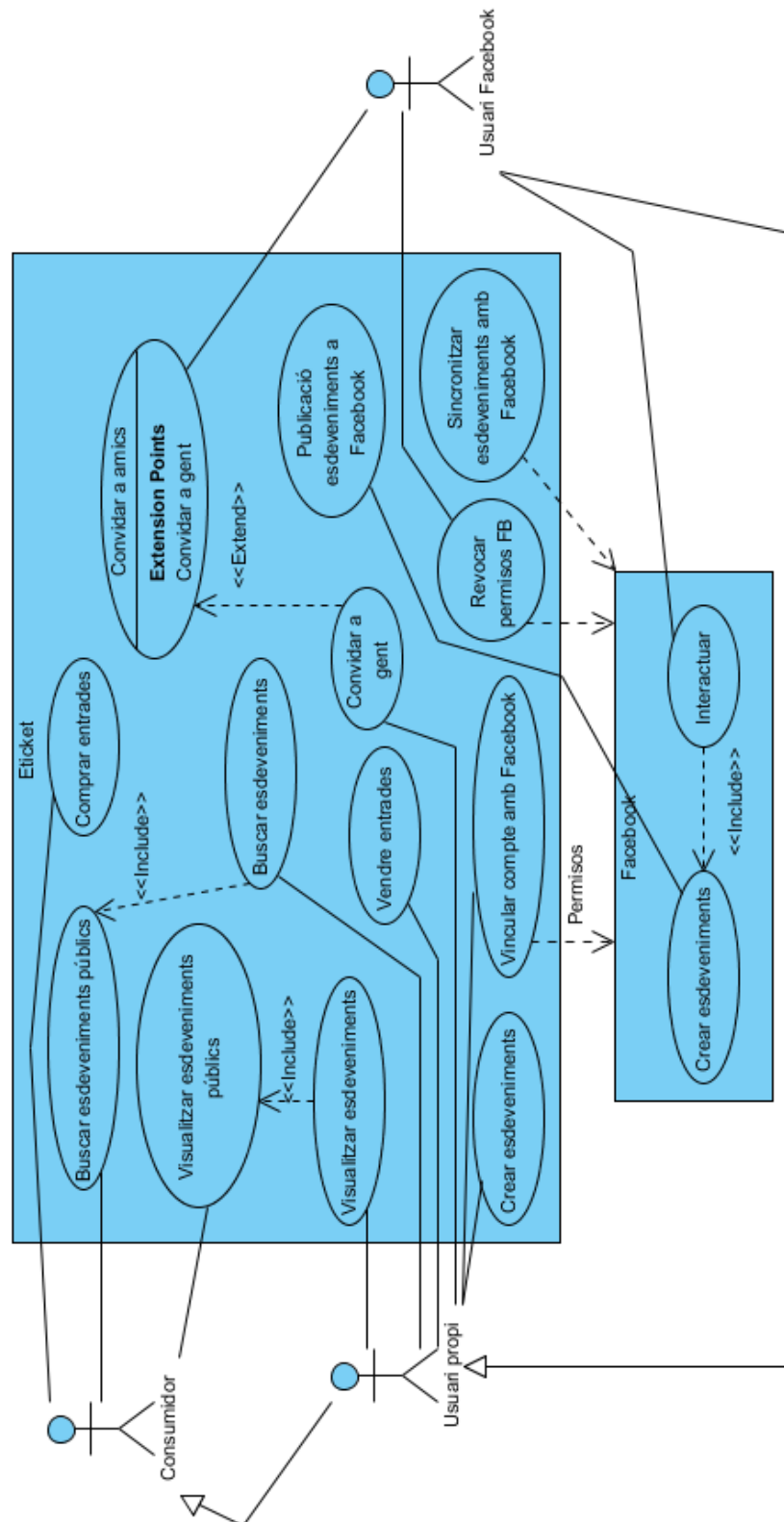


Figura 8. Diagrama de casos d'us

4.6. Llei Orgànica de Protecció de Dades

La Llei Orgànica de Protecció de Dades (LOPD) té per objecte garantir i protegir les llibertats públiques i els drets fonamentals de les persones físiques, en especial els de l'honor, intimitat i privacitat personal i familiar, pel que fa el tractament de dades de caràcter personal.

Regula el tractament que han de rebre els fitxers on es guarden dades esmentades anteriorment. La LOPD estableix tres nivells de seguretat:

- Nivell bàsic: qualsevol nivell que contingui dades personals
- Nivell mig: fitxer que contingui dades relatives a infraccions administratives o penals, serveis financers i solvència patrimonial i crèdit.
- Nivell alt: qualsevol fitxer que a més contingui dades sobre ideologia, religió, creences, origen racial, salut, vida sexual i dades amb finalitats policials sense el consentiment de les persones afectades.

Les dades que gestiona la nostra aplicació són de nivell bàsic i només hem de procedir a:

- Inscriure els fitxers de dades a l'Agència Espanyola de Protecció de Dades.
- Elaboració d'un document que recollirà les mesures d'índole tècnica i organitzativa d'acord amb la normativa de seguretat vigent que serà d'obligat compliment per al personal amb accés a les dades de caràcter personal. La AEPD proporciona una guia model per la realització del document:

https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicacion/es/common/Guias/modelo_doc_seguridad.doc

La sanció que estableix la llei davant les infraccions van dels 600 als 600.000€ en funció de la gravetat de la infracció.

5. Disseny

El següent capítol establirà el disseny que tindrà l'aplicació. El disseny és una part molt important i s'ha d'anar amb molt de compte ja que servirà de guia per implementar l'aplicació. S'han de tenir en compte tots els requeriments per tal d'oferir una bona interfície al client.

5.1. Estructura de la Base de Dades

5.2.1. Esquema de la BD

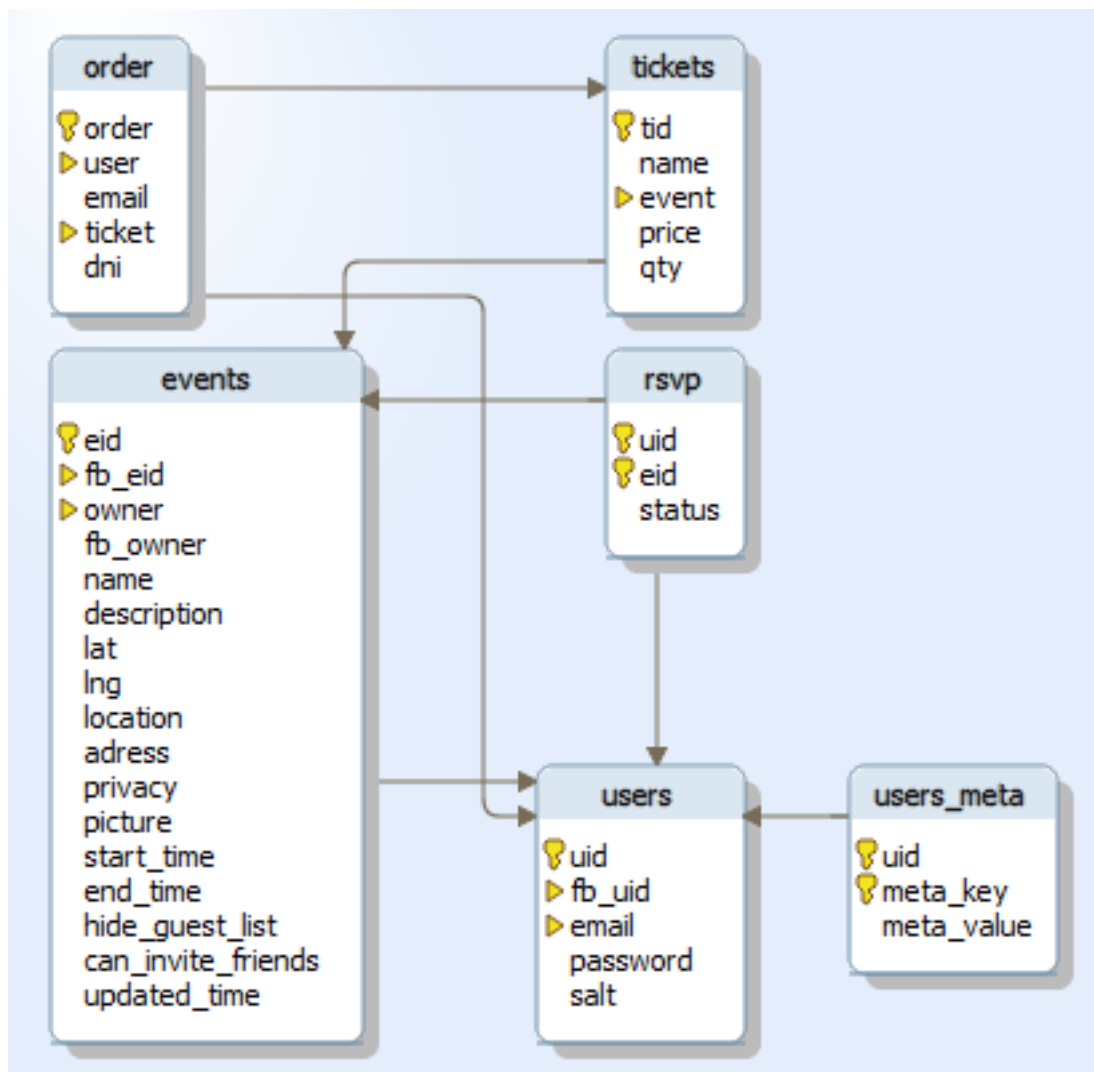


Figura 9. Esquema de la Base de Dades

5.1.2. Descripció de les taules

Taula users:

La taula users és l'encarregada de guardar les dades. Es guarden un identificador d'usuari, l'identificador de Facebook (si n'hi ha), l'e-mail, el hash de la contrasenya i el salt. El salt és una cadena de caràcters aleatòria que es genera durant el registre i s'afegeix a la contrasenya a l'hora d'encriptar-la i guardar-la per tal d'evitar atacs amb *rainbow tables* si per alguna raó la seguretat de la base de dades quedés compromesa.

Taula users_meta:

S'utilitzarà la taula users_meta per guardar dades temporals dels usuaris que no tinguin prou pes com per tenir una columna a la taula users. Un exemple ben clar és la clau per validar el compte de correu quan et registres.

Taula events:

És la taula principal on es guarda tota la informació relacionada amb els esdeveniments menys els tiquets i els convidats. Cada esdeveniment s'identifica per un ID únic. Els detalls inclosos són: l'ID de l'esdeveniment i del creador a Facebook (si és el cas), l'ID del creador si és usuari de l'aplicació, el títol, la descripció, la ubicació, opcions sobre la privacitat, una URL d'una imatge i l'horari.

Taula rsvp:

La taula rsvp (*Répondez s'il vous plaît*) conté la relació entre els usuaris i els esdeveniments com a convidats. Podem trobar l'identificador de l'usuari, de l'esdeveniment, i l'estatus de la resposta.

Taula tickets:

Aquesta taula conté els diferents tipus d'entrades que tindrà cada esdeveniment així com la quantitat d'entrades que queden disponibles. Inclou un nom per l'entrada i el preu.

Taula order:

Serà la taula encarregada d'emmagatzemar la informació sobre les compres d'entrades que es realitzin. Cada ordre de compra estarà relacionada amb un usuari. Si el comprador no està registrat es crearà un usuari a la taula sense només amb l'identificador d'usuari i la direcció de correu electrònic.

5.2. Estructura de l'aplicació

L'aplicació estarà dividida en diferents scripts que realitzaran les funcions analitzades anteriorment. Es poden classificar els mòduls en dos classes: accessibles i no accessibles per l'usuari. També existeixen certs mòduls que seran inaccessibles pels usuaris segons si han iniciat sessió o no.

5.2.1. Scripts externs: accessibles per l'usuari

Són els mòduls amb els que l'usuari interactuarà per utilitzar l'aplicació. Tenen una component gràfica i poden comunicar-se o accionar mòduls interns.

Entrada:

Serà la plana principal i permetrà a l'usuari anar a la resta de mòduls com iniciar sessió, registrar-se, buscar esdeveniments... Es veurà una llista amb els 15 esdeveniments més propers amb alguns detalls.

Registrar-se:

El mòdul per registrar-se permetrà als usuaris que no han iniciat sessió crear un compte per poder gaudir de les funcionalitats que ofereix l'aplicació. Serà necessari introduir una direcció de correu vàlida i una contrasenya segura. Quan s'introdueixin dades correctes el mòdul generarà el salt per la contrasenya i es comunicarà amb el de correu per enviar el link necessari per validar la direcció e-mail. No és accessible pels usuaris amb una sessió iniciada.

Validar compte:

Proporcionarà el servei per validar la direcció de correu electrònic mitjançant una clau que es generarà en el moment de registrar-se. Quan es finalitzi el procés l'usuari podrà iniciar sessió amb les seves credencials.

Iniciar sessió:

Serà l'encarregat d'iniciar la sessió de l'usuari. Per iniciar sessió farà falta introduir la direcció de correu i la contrasenya triada a l'hora de registrar-se. Quan un usuari iniciï sessió amb Facebook es cridarà al mòdul recol·lector per importar tots els esdeveniments de Facebook que ell pugui veure a través seu o dels seus amics. Serà inaccessible pels usuaris amb una sessió iniciada.

Tancar sessió:

S'ocuparà de tancar la sessió del usuari ja sigui pròpia o amb Facebook. Immediatament redirigirà l'usuari al mòdul d'entrada.

Buscar esdeveniment:

Buscarà els esdeveniments segons els detalls facilitats i els mostrarà en pantalla. Només mostrarà els esdeveniments públics i aquells on estiguis convidat, per tant, un usuari sense sessió només veurà esdeveniments oberts.

Crear esdeveniment:

Aquest mòdul permetrà als usuaris registrats crear esdeveniments introduint els seus detalls. Si s'activa la opció corresponent efectuarà les operacions necessàries per crear l'esdeveniment també a Facebook. Introduirà les dades a la base de dades un cop finalitzat el procés. Només serà accessible pels usuaris amb una sessió iniciada.

Editar/eliminar esdeveniment:

Servirà per editar o eliminar els esdeveniments creats. Mostrarà una interfície per poder canviar els detalls i s'encarregarà d'actualitzar la base de dades. Només serà accessible pels usuaris amb una sessió iniciada.

Visualitzar esdeveniment:

L'script mostrarà visualment els detalls del esdeveniment. Permetrà interactuar amb altres scripts com el de convidar, o comprar entrades. Si l'esdeveniment està a Facebook les dades s'agafaran directament d'allà per garantir la sincronització. Serà accessible per tots els usuaris però els esdeveniments privats només podran visualitzar-se pels seus convidats.

Comprar entrades:

Processarà la compra i el mètode de pagament. Un cop realitzat la venda amb èxit, procedirà a inserir les dades de la comanda a la base de dades i activarà l'script de correu per a que envii un correu de confirmació.

Convidar:

Efectuarà les connexions necessàries amb el mòdul de correu o amb Facebook per tramitar les invitacions a l'esdeveniment. També processarà les respostes a les invitacions i introduirà les dades necessàries a la base de dades.

5.2.2. Scripts interns: inaccessibles per l'usuari

Correu:

Enviarà correus electrònics als usuaris per notificar invitacions, validacions de comptes o confirmacions de compres efectuades correctament.

Recol·lector:

S'activarà cada cop que un usuari de Facebook inici sessió i recopilarà els esdeveniments que pot accedir l'usuari i els que els seus amics comparteixen. Si els esdeveniments ja es troben a la base de dades no seran inserits.

5.2.3. Accessibilitat dels scripts segons l'usuari

A continuació es presentarà una taula amb els script accessibles pels usuaris i és marcarà quins seran els usuaris que podran accedir i quins no.

Script	Usuari sense sessió	Usuari amb sessió iniciada
Entrada	X	X
Registrar-se	X	
Validar compte	X	
Iniciar sessió	X	
Tancar sessió		X
Buscar esdeveniment	X*	X
Crear esdeveniment		X
Editar/eliminar esdeveniment		X
Visualitzar esdeveniment	X*	X
Comprar entrades	X	X
Convidar		X

Taula 14. Accessibilitat dels scripts segons l'usuari

Les X amb asterisc significa que són accessibles però només parcialment, sense totes les funcionalitats.

5.3. Disseny de la interfície d'usuari

El disseny de la interfície gràfica és potser la part més important de cara a l'usuari. El disseny afecta a la quantitat d'esforç que ha de dedicar l'usuari per aportar i extreure informació al sistema i al temps que necessita per aprendre a fer-ho. Una interfície d'usuari mal feta pot arruïnar tota una aplicació encara que disposi de bones funcionalitats. El nostre objectiu serà la realització d'una interfície gràfica de qualitat i.e. accessible i intuïtiva i amb gran usabilitat.

5.3.1. Elements bàsics de la interfície.

Per a realitzar la interfície d'usuari s'empraran elements de la llibreria jQuery UI i els seus fulls d'estil. El tema (full d'estil) triat pels *widgets* ha estat el South Street per la seva aparença neta i professional combinada amb un color verd vistós. Els *widgets* que s'utilitzaran seran:

- **Dialogs:** finestres emergents que es situen per sobre la visualització de la pàgina i permeten mostrar missatges o formularis.



The image shows a jQuery UI Dialog box titled "Crea tu cuenta en Eticket". It contains three text input fields labeled "Email", "Contraseña", and "Repita la Contraseña". At the bottom right, there are two buttons: "Registrarse" and "Cancelar".

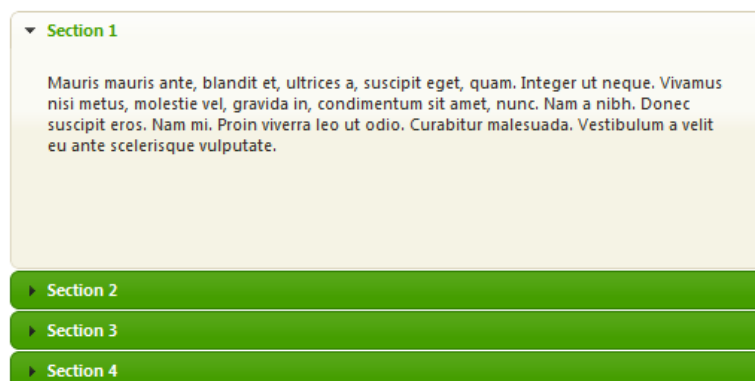
Figura 10. UI Dialog

- **Botons:** Botons per interactuar amb l'aplicació.
- **Calendari:** Calendari per introduir dates a formularis.



Figura 11. UI Calendar

- **Acordiό:** Mostra informació i només permet tenir una secció oberta.



The image shows a jQuery UI Accordion widget. It has four sections: "Section 1", "Section 2", "Section 3", and "Section 4". "Section 1" is expanded, showing a paragraph of placeholder text. The other sections are collapsed.

Figura 12. UI Accordion

- **Autocompletar:** Element per autocompletar les cerques i poder escollir les alternatives d'una llista.

5.3.2. Interfície base

La interfície de la web constarà dels següents elements:

- **Capçalera:** Es situarà a dalt de tot de la pàgina i contindrà el logo, els botons per iniciar sessió i registrar-se, el menú i la barra de cerca. Quan l'usuari hagi iniciat sessió en comptes dels botons es mostrarà la direcció de correu o el nom de l'usuari.
- **Cos:** Aquesta parta serà la que canviï segons la pàgina on estiguem. Encara que canviï mantindrà el fons sempre.
- **Peu de pàgina:** Estarà format per una línia horitzontal divisòria i el nom del autor i el projecte.



Figura 13. Interfície base

El cos s'omplirà amb la informació que es requereixi en cada moment fent ús dels elements esmentats a la secció anterior. Per exemple:

Titulo del evento muy largo son 75 caracteres para el ejemplo

Asistiré
Quizá artista
No asistiré

31

Tickets

Adulto	21.45 €	Quedan 100 entradas	Comprar ahora
Niño	15.00 €	Quedan 68 entradas	Comprar ahora

Figura 14. Exemple cos UI

6. Implementació

La implementació és la part més extensa del projecte i és on es posa en pràctica tot l'anàlisi i disseny anteriorment realitzats per materialitzar-ho en una aplicació que haurà de complir amb els requeriments proposats. En aquesta part es on t'adones dels errors comesos durant l'anàlisi i el disseny i has de recórrer a altres vies per tal de solucionar els problemes que vagin sortint.

6.1. Servidor WAMP

Com que es va decidir utilitzar les tecnologies Apache, MySQL i PHP, és va triar el servidor WAMP per a allotjar l'aplicació. El servidor WAMP (*Windows, Apache, MySQL & PHP*) és una aplicació que et permet posar en marxa un servidor amb totes aquestes tecnologies molt fàcilment i sense haver de perdre temps configurant les aplicacions per a què puguin interactuar entre elles. Això va permetre estalviar una mica de temps per poder-lo dedicar a solucionar altres problemes que varen anar sortint. La versió del servidor WAMP és la 2.1a (l'última) i inclou Apache 2.2.17, Php 5.3.3, Mysql 5.5.8 i PhpMyadmin 3.2.0.1. PhpMyAdmin és una aplicació web molt interessant per gestionar la base de dades gràficament en comptes de utilitzar comandes de la consola.

6.2. Implementació de l'aplicació

6.2.1. Base dels scripts externs PHP

Tots els scripts externs contenen l'script base.php, que conté una col·lecció de funcions i inicialitza certes variables necessàries per la correcta visualització de les pàgines en els casos que no són per defecte e.g. quan un usuari es registra i se li ha de comunicar que rebrà un e-mail amb el link per activar el seu compte.

Funcions que conté:

- **base64UrlEncode/Decode:** per codificar i ofuscar dades i enviar-les com a paràmetres a les URL.
- **isLoggedIn:** Aquesta funció és l'encarregada d'establir si l'usuari té una sessió iniciada i d'inicialitzar les variables necessàries si és el cas. Retorna un número en funció de si està connectat amb un compte propi o de Facebook, ja que la visualització de les webs pot canviar segons el cas.

L'script també inicialitza les variables `$script` i `$script_fb` que s'utilitzaran als diferents scripts per afegir codi JS en funció del flux que segueixi el codi PHP.

6.2.2. Connexió a la base de dades i la extensió `mysqli`

La connexió a la base de dades està disposada en un script a part per facilitar una possible modificació dels credentials de la connexió. Des dels scripts que es necessiti cap funció de la base de dades es cridarà la funció `require('db_connection.php')` per disposar de la connexió a la variable `$db`.

Per realitzar les operacions i.e. les *queries*, s'utilitzarà l'extensió `mysqli` per PHP. `mysqli` vol dir MySQL Improved i ve a substituir l'antiga extensió `mysql`. Com a novetat principal introdueix els *prepared statements* que són *queries* que es passen al *parser* del servidor MySQL amb uns interrogants sobre les variables que aniran canviant. A part de l'increment de velocitat causat per no haver d'analitzar la *query* cada vegada que s'executa, introdueix una millora importantíssima en seguretat pel que fa els atacs de *SQL Injection*. Com que la consulta ja haurà estat analitzada pel servidor, les variables que després ompliran els interrogants seran tractades únicament com variables per comparar o agafar el seu valor impeding totalment els atacs d'injecció SQL ja que no té en compte caràcters especials perquè la consulta ja ha estat *parsejada*. Sempre que faci falta utilitzarem els *prepared statements* per estalviar-nos feina a l'hora de comprovar les dades introduïdes per l'usuari i alhora millorar el nivell de seguretat de l'aplicació.

6.2.3. Sessions PHP

Les sessions PHP permeten mantenir dades entre scripts i així poder mantenir informació de si s'ha iniciat sessió o altres dades d'interès. Les sessions de PHP utilitzen un identificador de sessió i aquest és pot transmetre a través de les URL o amb una cookie. S'ha configurat el servidor per a que utilitzi les sessions només via cookie ja que transmetre l'identificador per les URL comporta molts problemes de seguretat. Tot i així les sessions són vulnerables a atacs XSS. El que s'emmagatzema a la cookie però, és només l'identificador i les dades de la sessió es guarden al servidor. Inclourem `session_start()` al començament dels nostres scripts per permetre'ns utilitzar les dades de la sessió. Per accedir-hi s'utilitza la superglobal `$_SESSION[]`, un array on és pot guardar qualsevol tipus de dades encara que no es recomana posar-hi dades importants com contrasenyes.

6.2.4. Sistema per registrar-se i iniciar sessió

El sistema per registrar-se exigeix una contrasenya de com a mínim 7 caràcters i una direcció de correu electrònic vàlida. Per comprovar si la direcció té un format correcte s'utilitza una expressió regular. Un cop realitzat el procediment, s'envia un correu a la direcció amb un link al script per activar que conté una clau aleatòria (el hash SHA-1 del salt) que és guarda a la taula de users_meta juntament amb una data d'expiració per l'activació. Quan l'script activate.php rep les variables de la URL verifica la clau i introdueix el salt a la taula de users per indicar que l'usuari ja està actiu.

El sistema d'iniciar sessió, com hem explicat més enrere, s'utilitza un *salt* per emmagatzemar les contrasenyes xifrades. La utilització del salt millora substancialment la seguretat de les nostres contrasenyes en cas que algun intrús pogués accedir a la nostre base de dades. Afegint una cadena de caràcters aleatòria al final de la contrasenya abans d'encriptar-la reduïm moltíssim la probabilitat de que els atacs de *rainbow table* (taules amb col·leccions immenses de paraules i el seu hash corresponent) tinguin èxit. La funció hash utilitzada per protegir les contrasenyes és SHA-256 que ens dona una sortida de 64 caràcters hexadecimal. Quan s'han verificat les credencials de l'usuari es crea una variable a la sessió PHP amb el seu identificador i correu electrònic. Si l'usuari ja ha iniciat sessió, els scripts redirigeixen al client al index mitjançant la funció *header()*.

6.2.5. Script per enviar e-mails

A l'hora d'enviar correus es van estudiar diverses opcions com posar en marxa un servidor. Tot i així, per comoditat es va decidir crear un compte de GMail (eticketpfc@gmail.com). La connexió i l'enviament del correu és fa mitjançant la llibreria PHPMailer.

6.2.6. Tancar sessió

L'script per tancar sessió simplement s'encarrega de destruir la sessió esborrant les variables i les cookies de la sessió PHP i la sessió de Facebook (més endavant es veurà). Quan ha acabat redirigeix a l'usuari al index.

6.2.7. Tasques periòdiques

Per realitzar tasques periòdiques com l'eliminació dels usuaris no validats s'utilitza la utilitat nnCron LITE que realitza peticions al servidor mitjançant CURL periòdicament segons se li especifiqui.

6.2.8. Nom de domini

Per a poder realitzar la implementació amb Facebook es requeria un nom de domini i es va decidir utilitzar el servei gratuït que ofereix www.no-ip.com per crear subdominis del tipus <http://subdomini.no-ip.org> i crear el nostre propi domini <http://eticket.no-ip.org> que s'actualitza automàticament encara que tinguis una IP dinàmica mitjançant una lleugera aplicació.

6.2.8. URL amigables

Avui dia és molt important disposar de URL simples i clares per tal de facilitar la vida als buscadors i als usuaris. Tot i així de vegades és impossible si s'han de passar paràmetres amb GET a un determinat fitxer. És per aquesta raó que Apache disposa d'un mòdul, *mod_rewrite*, que facilita la transformació de URLs. Per exemple podem transformar una URL del tipus <http://eticket.no-ip.org?events.php?eid=1234> en <http://eticket.no-ip.org/events/1234>. La transformació es realitza mitjançant expressions regulars com per exemple:

```
RewriteRule ^events(?:/([0-9]*))?$ /events.php?eid=$1
RewriteRule ^login/(?:\?([A-Za-z0-9&=_;\.\@\+]*))?$ login.php?$1
```

Les URL amigables són molt importants a l'hora de buscar un bon posicionament als buscadors.

6.3. Implementació de la interfície d'usuari

Per implementar la interfície s'utilitzen els fitxers CSS (fulls d'estil) juntament amb JS, jQuery i jQuery UI. Totes les pàgines tenen al començament del body un *require('header.php')* i un *require('footer.php')* al final d'aquest. Aquesta tècnica ens facilita la modificació de la capçalera i el peu de pàgina.

Els fitxers que s'inclouen a la capçalera HTML són:

CSS:

<http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/themes/south-street/jquery-ui.css>

<http://eticket.no-ip.org/styles/basic.css>

JS:

<https://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.js>

<https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/jquery-ui.js>

<http://eticket.no-ip.org/lib/basic.js>

Tal com es pot veure s'utilitzen les versions allotjades a Google de les llibreries JS i el CSS de jQuery UI per la velocitat que garanteix aquest i per alliberar de càrrega al nostre servidor.

6.3.1. Script bàsic

L'script basic.js s'inclou a totes les webs i permet conté les funcions per crear els widgets de jQuery UI així com per validar camps de formularis. Podem trobar les funcions:

- **checkEmpty:** Comprova que tots els camps del formulari estiguin plens.
- **checkRegexp:** Comprova si el camp desitjat compleix una expressió regular determinada. Serveix principalment per comprovar emails.
- **updateTips:** És l'encarregada de mostrar els missatges d'alerta quan hi ha algun camp incorrecte als formularis. Les dues primeres funcions criden a aquesta si detecten errors als camps.
- **toggleInputPrefilled:** Fa que el text per defecte d'alguns camps com el de cerca aparegui i desaparegui.

6.3.2. Validació dels formularis

Seguidament veurem un exemple de validació de formularis mitjançant les funcions esmentades en la subsecció anterior.

```
var email = $( "#email_reg" ),
    pass1 = $( "#pass1" ),
    pass2 = $( "#pass2" ),
    allFields = $( [] ).add( email ).add( pass1 ).add( pass2 );
allFields.removeClass("ui-state-error");
$( '[id$="_tip"]' ).text('').hide().prev().hide();
var valid = true;
valid = checkEmpty(allFields) ? valid : false;
valid = email.val().length != 0 && checkRegexp(email,
| /^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,6}$/i,
    'El correo no tiene un formato válido') ? valid : false;
if (pass1.val() != pass2.val()){
    valid = false;
    updateTips(pass1, 'Las contraseñas introducidas no son iguales');
    pass1.val('');
    pass2.val('').addClass('ui-state-error');
}
```

Figura 15. Exemple de codi per validar

I aquí podem veure el que seria una validació errònia:

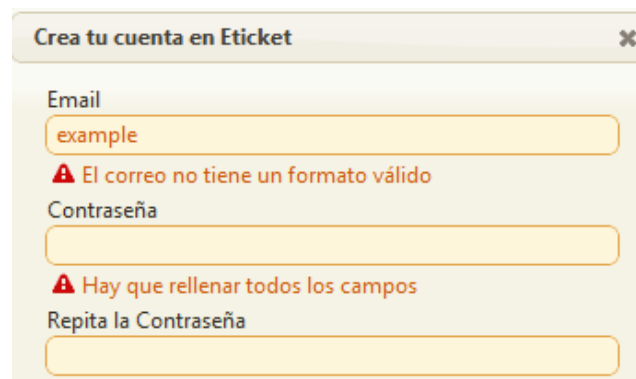


Figura 16. Exemple formulari no vàlid

Tot i que es verifiquin els camps mitjançant JS mai podem deixar aquesta validació com única i sempre s'haurà de comprovar el valor correcte dels camps abans d'introduir-los al sistema amb PHP.

6.3.3. Estil de finestres

Tot el disseny de la interfície està fet mitjançant finestres per tal de facilitar la usabilitat de l'aplicació. Les cantonades estan arrodonides per aconseguir una visió més moderna. Existeix però un problema amb les cantonades arrodonides: la propietat només està disponible a CSS 3 i hi ha navegadors que encara no implementen l'estàndard. Tot i així s'ha decidit mantenir l'estil tot i el risc de no

complir els requisits no funcionals ja que s'ha argumentat que no implica una diferència en la usabilitat.



Próximos eventos			
	CUMPLEAÑOS MANU (PUZZLE) EN SOL I LLUNA (XERACO)	SOL I LLUNA CHIRINGUITO	19/09 @ 00:00h €
	☆ MISH:MASH EVENTS ☆ STUDENT RECRUITMENT 2011 ☆ GET PAID TO PARTY! ☆	England	19/09 @ 00:00h €
	The *OFFICIAL* Hallam Union Welcome Week	The Hubs	19/09 @ 00:00h €
	Liga AZUL 2011 II Fase	Practicapadel Alaquàs	19/09 @ 01:00h €
	La aventura del Guerrero: Descubre el poder que hay en ti!	Desierto de Marruecos	19/09 @ 01:00h €
	"R. Zaragoza - RCD ESPAÑOL".	La Romareda.	19/09 @ 01:00h €

Figura 17. Exemple disseny finestres

6.3.4. Capçalera

La capçalera canvia segons si haguem iniciat o no:



Figura 18. Capçalera amb i sense sessió iniciada

6.4. Implementació de la connexió amb Facebook

Facebook proporciona una sèrie de APIs per interactuar amb la seva xarxa: la Graph API, Rest API (en vies d'extinció) i FQL (**F**acebook **Q**uery **L**anguage) així com tota una sèrie de plugins ja desenvolupats i llestos per afegir a la teva aplicació. Per interactuar amb les API també ofereix SDK per JS i PHP que ens seran molt útils per facilitar-nos la feina.

6.4.1. APIs

Graph API:

És tracta de la nova API de Facebook que sustituirà a la Rest API quan s'hagin migrat totes les funcionalitats. Ofereix una facilitat d'ús increïble ja que només s'han d'utilitzar URL i ids per accedir als recursos. Els recursos són tornats en format JSON i cada recurs està connectat als altres formant un gran graf. Per exemple per accedir a la informació del meu perfil simplement s'ha d'accedir a <https://graph.facebook.com/1067993517> i per accedir als meus amics <https://graph.facebook.com/1067993517/friends> (necessari access_token, més endavant s'explica). No només permet fer consultes sinó que també es poden modificar dades, crear-ne o esborrar-ne mitjançant els mètodes de petició d'HTTP POST i DELETE. Com a inconvenients però tenim que encara no té totes les funcions migrades i presenta alguns *bugs* degut a la seva curta vida. També

tenim el problema de la documentació que aporta Facebook ja que degut a la constant actualització de la API de vegades està una mica desfasada.

Rest API:

És la API que hi havia fins que van treure la Graph API. Permet realitzar qualsevol interacció amb Facebook i està àmpliament provada. Facebook no recomana el seu ús ja que poc a poc es va migrant totes les funcions i acabarà sent *deprecated*.

FQL:

FQL ofereix una interfície per accedir a les dades de Facebook com si estiguéssim realitzant consultes a una base de dades amb un llenguatge una mica més restringit. És molt més potent que la Graph API a l'hora de recollir una gran quantitat de dades (és capaç d'obtenir informació de fins a 2500 esdeveniments d'un cop). Les dades es divideixen en taules i per realitzar les consultes s'ha d'utilitzar un dels camps obligatoris a les condicions. No podem realitzar consultes de l'estil `SELECT * FROM event` o `SELECT eid, title FROM event WHERE location = 'Blanes'` ja que el camp obligatori per la taula event és l'eid. Per realitzar una query simplement s'ha de demanar un recurs via URL: [https://api.facebook.com/method/fql.query?query="query"](https://api.facebook.com/method/fql.query?query=\). La resposta s'obté en XML per defecte però també es pot obtenir en JSON afegint `format=json` a la URL. L'únic inconvenient és que només ens permet fer consultes.

6.4.2. Autenticació

Per poder utilitzar les APIs s'ha de crear una aplicació i obtenir l'App ID i l'App Secret. Aquests dos números seran els que s'utilitzaran perquè els usuaris iniciïn sessió a la teva web i per poder obtenir la access token. Facebook ha migrat recentment a OAuth 2.0, un protocol obert que permet l'autorització segura, senzilla i estàndard d'una API. Quan l'usuari autoritza la teva aplicació obtens un access token que et permet realitzar accions en nombre seu (depenent dels permisos que t'hagi concedit). Les access tokens caduquen en un temps curt per garantir un mínim de seguretat si l'access token cau en males mans.

6.4.3. SDKs

Per realitzar el l'autenticació i les trucades a les APIs Facebook proporciona uns SDKs per PHP i JS que faciliten gratament la feina. Per a poder utilitzar tots dos SDKs conjuntament el procés d'autenticació s'ha de realitzar amb el JS SDK amb la opció cookie activada perquè el PHP SDK pugui accedir-hi i obtenir les dades

de la sessió. Aquí van sorgir els primer problemes perquè Facebook està en ple procés de migració a OAuth 2.0 (acaba aquest mes de setembre) i la cookie que utilitza el JS SDK va canviar i el PHP SDK era incapaç d'obtenir la sessió fins que el van actualitzar i me'n vaig adonar. El següent problema va sorgir per la compatibilitat entre navegadors. En la aplicació, l'autenticació es fa mitjançant JS SDK i PHP SDK recull la informació de la cookie a la pàgina següent però Firefox no crea la cookie de Facebook a la pàgina següent que es visiti sinó a la que va després. Això va obligar a fer unes redireccions invisibles per l'usuari mitjançant la funció *header()* de PHP.

JS SDK:

Facilita la interacció amb Facebook mitjançant l'objecte FB, que s'ha d'inicialitzar, i els seus mètodes.

PHP SDK:

Proporciona la interacció amb Facebook aprofitant l'orientació a objectes introduïda a PHP 5.

Intentarem utilitzar sempre que puguem el JS SDK ja que la informació es demana asíncronament i des de l'ordinador del client de manera que evitem carregar el servidor amb peticions evitables. Només utilitzarem el PHP SDK per processos en segon pla on no es pugui utilitzar el JS SDK.

6.4.5. Permisos

Facebook exigeix permisos per part de l'usuari per realitzar accions i accedir a certa informació. Els permisos que la aplicació exigeix com a bàsics són *user_events* i *friends_events* per poder recol·lectar els esdeveniments de l'usuari i els que pugui veure dels seus amics. Després hi ha una sèrie d'esdeveniments complementaris segons les accions que es vulguin realitzar dins de l'aplicació:

- ***create_event***: Permet a l'aplicació crear i modificar esdeveniments en nom de l'usuari. De moment existeix un bug reportat en la Graph API i només se't permet modificar aquells esdeveniments que hagi creat l'aplicació a través de la Grap API.
- ***rsvp_event***: Habilita a l'aplicació per respondre a les invitacions en nom de l'usuari.
- ***read_stream***: Proveeix d'accés a tots els posts del News Feed. L'aplicació l'exigeix per publicar el mur de l'esdeveniment a l'aplicació.

6.4.4. Recol·lector

El collector.php és l'encarregat d'importar els esdeveniments propis i dels amics a l'aplicació. És un script que s'executa en segon pla i quan l'usuari inicia sessió a Facebook. Utilitza l'acces token que se li passi per la URL i no és accessible pels usuaris externs. És capaç d'inserir 2500 esdeveniments per cada petició que fa a Facebook. Si es demanen més esdeveniments via FQL es corre el risc de rebre un error desconegut per part de Facebook probablement degut a la dimensió de la consulta. Ha de realitzar algunes adaptacions de les dades abans d'introduir-les a la base de dades com transformar els caràcters especials a HTML o transformar l'adreça al format correcte.

7. Proves

7.1. Proves realitzades

S'han anat realitzant proves conforme s'anava programant tota l'aplicació per veure si el funcionament era el desitjat però tot i així s'han realitzat més proves al final. Concretament s'ha realitzat:

- S'ha comprovat que el recol·lector podia assimilar una gran quantitat d'esdeveniments sense col·lapsar.
- S'ha provat d'introduir valors invàlids als formularis inclús realitzant atacs d'injecció HTML per veure si el codi PHP estava ben dissenyat.
- S'han realitzat atacs de SQL Injection per veure com es comportava el codi.
- S'han imprès al log d'errors variables intermèdies per comprovar que tenien els valors correctes.
- S'ha comprovat si el mòdul de correu era capaç d'enviar varis correus simultàniament.
- S'han realitzat proves de compatibilitat *cross-browser* amb èxit exceptuant la visualització arrodonida de les cantonades als navegadors sense suport per CSS 3.
- S'han realitzat accions incongruents per veure la resposta de l'aplicació com intentar iniciar sessió de compte propi un cop estava la sessió de Facebook iniciada.

8. Conclusions

8.1. Conclusions finals

La raó principal del projecte final de carrera és la d'aprendre a desenvolupar individualment un projecte com els que ens podem trobar al món laboral on hem d'identificar el problema i seguir tots els passos necessaris abans d'afrontar-lo com són el l'anàlisi i el disseny. Hem d'avaluar si el projecte és viable i defensar-lo públicament. Puc treure les següents conclusions de la realització del projecte:

- És una eina fabulosa per aprendre a buscar-te la vida tu mateix ja que les pràctiques durant la carrera solen ser molt pautades i no hi ha gaires maneres de solucionar un mateix problema.
- Serveix per discernir entre els que treballen i els que no ja que al llarg de tota la carrera es fan totes les coses en grup i de vegades no és possible establir si algú ha treballat o no.
- S'aprèn moltíssim de la realització d'un projecte des de l'inici fins al final ja que vius totes les fases d'un mateix projecte i veus com totes les peces del procés van encaixant per acabar desenvolupant una aplicació que soluciona algun tipus de problema.
- És molt important realitzar un bon anàlisi i disseny per fer-ne via quan comences a implementar ja que si tens les idees clares la implementació es fa molt més amena ràpida i menys errònia.
- És de vital importància no deixar les coses pel final perquè després et poc quedar un gust agredolç per no haver pogut incloure tot el que t'hagués agradat a més de passar-ho malament el últims dies per no saber si acabaràs.

Per finalitzar m'agradaria comentar que vaig triar aquest projecte perquè a mi no m'agradava massa la programació web però creia que en el fons podia estar bé i em podia obrir moltes portes al futur de cara a muntar un negoci per mi mateix o al món laboral. Després d'haver aprofundit en aquest món puc concloure que em porto una bona experiència i m'ha agradat més del que em pensava en un principi.

8.2. Millores i ampliacions

Hi ha un infinit nombre d'ampliacions possibles que no s'han pogut implementar per falta de temps o per complexitat massa elevada però que poden ajudar a completar l'aplicació i fer-la més útil.

- Poder exportar les llistes de convidats a altres formats com Excel o PDF.
- Poder crear codis de descompte per les entrades.
- Fer que les entrades es puguin imprimir.
- Incloure un codi de barres en les entrades per poder llegir-lo amb un lector a l'entrada de l'esdeveniment.
- Desenvolupar aplicacions per Android o iPhone per poder realitzar lectures de codis de barres o QR.
- Poder publicar els esdeveniments en altres xarxes socials com LinkedIn.
- Introduir comentaris als esdeveniments.
- Que a més de mostrar el mur de Facebook als esdeveniments també es pugui interactuar amb ell i publicar.
- Crear una aplicació Canvas per a poder comprar les entrades directament des de Facebook.
- Elaboració d'un algoritme per posar els esdeveniments més importants a la plana principal.
- Que els usuaris tinguin una pàgina pròpia i altres usuaris puguin subscriure's.
- Que es desenvolupi una interfície per poder fer el check-in amb un ordinador/pad a la porta de l'esdeveniment.
- Crear un sistema de carro de la compra per no haver de comprar les entrades d'una en una.
- Introduir tooltips per mostrar explicacions i consells als elements de l'aplicació.

Aquestes són algunes millores que se m'han anat acudint durant la elaboració del projecte i crec que l'aplicació té molt potencial per fer-se més gran, interactiva i social. Crec que algunes d'aquestes millores poden convertir-se en projectes en el futur.

9. Bibliografia

9.1. Llibres consultats

- Luke Welling, Laura Thomson. **Desarrollo web con PHP y MySQL**. Madrid: Anaya Multimedia, cop. 2009. ISBN 9788441525535
- Kabir, Mohammed J. **La Biblia de servidor Apache 2**. Madrid : Anaya Multimedia, cop. 2002. ISBN 8441514682
- Pérez López, César. **Dreamweaver 8 : Desarrollo de páginas web dinámicas con PHP y MySQL**. Madrid : Ra-Ma, cop. 2007 ISBN 978-84-7897-752-9

9.2. Pàgines web consultades

- **JLBN - Free Wamp Guides & Web Designs Templates** (Accés 8/11)
<http://jlbn.net/>
- **Tinsology** (Accés 8/11)
<http://tinsology.net/>
- **PHP SDK 3.0 & Graph API base Facebook Connect Tutorial** (Accés 9/11)
<http://thinkdiff.net/facebook/php-sdk-3-0-graph-api-base-facebook-connect-tutorial/>
- **Facebook Wall - a jQuery Social Media Plugin** (Accés 9/11)
<http://www.neosmart.de/social-media/facebook-wall>
- **jQuery UI - Demos & Documentation** (Accés 9/11)
<http://jqueryui.com/demos/>
- **jQuery JavaScript Library – Documentation** (Accés 9/11)
<http://docs.jquery.com/>
- **PHP: Hypertext Preprocessor Manual** (Accés 9/11)
<http://php.net/>

- **PayPal Dev Central Home Page** (Accés 9/11)
- https://cms.paypal.com/es/cgi-bin/?cmd=_render-content&content_ID=developer/home&nav=3
- **Facebook Developers** (Accés 9/11)
- <https://developers.facebook.com/>
- **Stackoverflow** (Accés 9/11)
- <http://stackoverflow.com/>
- **Google Maps Javascript API V3 – Google Code** (Accés 9/11)
- <http://code.google.com/apis/maps/documentation/javascript/>
- **Introduction To MySQLi And Prepared Statements - PHP Tutorials** (Accés 8/11)
- <http://www.dreamincode.net/forums/topic/54239-introduction-to-mysqli-and-prepared-statements/>

10. Apèndix

10.1. Índex de figures

Figura 1. Lògica del sistema actual	8
Figura 2. Tasques del projecte detallades.....	13
Figura 3. Diagrama de Gantt previst	13
Figura 4. Esquema de generació dinàmica de continguts	19
Figura 5. Esquema de funcionament del model AJAX.....	20
Figura 6. Widgets jQuery UI.....	21
Figura 7. Esquema del protocol HTTPS	24
Figura 8. Diagrama de casos d'us	27
Figura 9. Esquema de la Base de Dades.....	29
Figura 10. UI Dialog	34
Figura 11. UI Calendar	34
Figura 12. UI Accordion	34
Figura 13. Interfície base	35
Figura 14. Exemple cos UI.....	35
Figura 15. Exemple de codi per validar	41
Figura 16. Exemple formulari no vàlid	41
Figura 17. Exemple disseny finestres.....	42
Figura 18. Capçalera amb i sense sessió iniciada	42

10.2. Índex de taules

Taula 1. Prioritat dels objectius del projecte	5
Taula 2. Stakeholders	6
Taula 3. Perfils d'usuari	6
Taula 4. Project Team.....	7
Taula 5. Usuaris i/o personal del sistema	8
Taula 6. Catalogació i priorització dels requisits	10
Taula 7. Relació entre els requisits i els objectius	10
Taula 8. Comparació d'alternatives	12
Taula 9. Recursos del projecte.....	12
Taula 10. Riscos del projecte	14
Taula 11. Plans de contingència	14
Taula 12. Estimació cost de personal.....	15
Taula 13. Estimació cost dels recursos.....	15
Taula 14. Accessibilitat dels scripts segons l'usuari	33